DSC 461 Project - Job Application Database

Milestone 2

Team: Fastest HK Journalist (# 11)

Members: Zhou Xu (zxu17), Kefu Zhu (kzhu6), Hao Jiang (hjiang23)

Task A: Draw an ER diagram

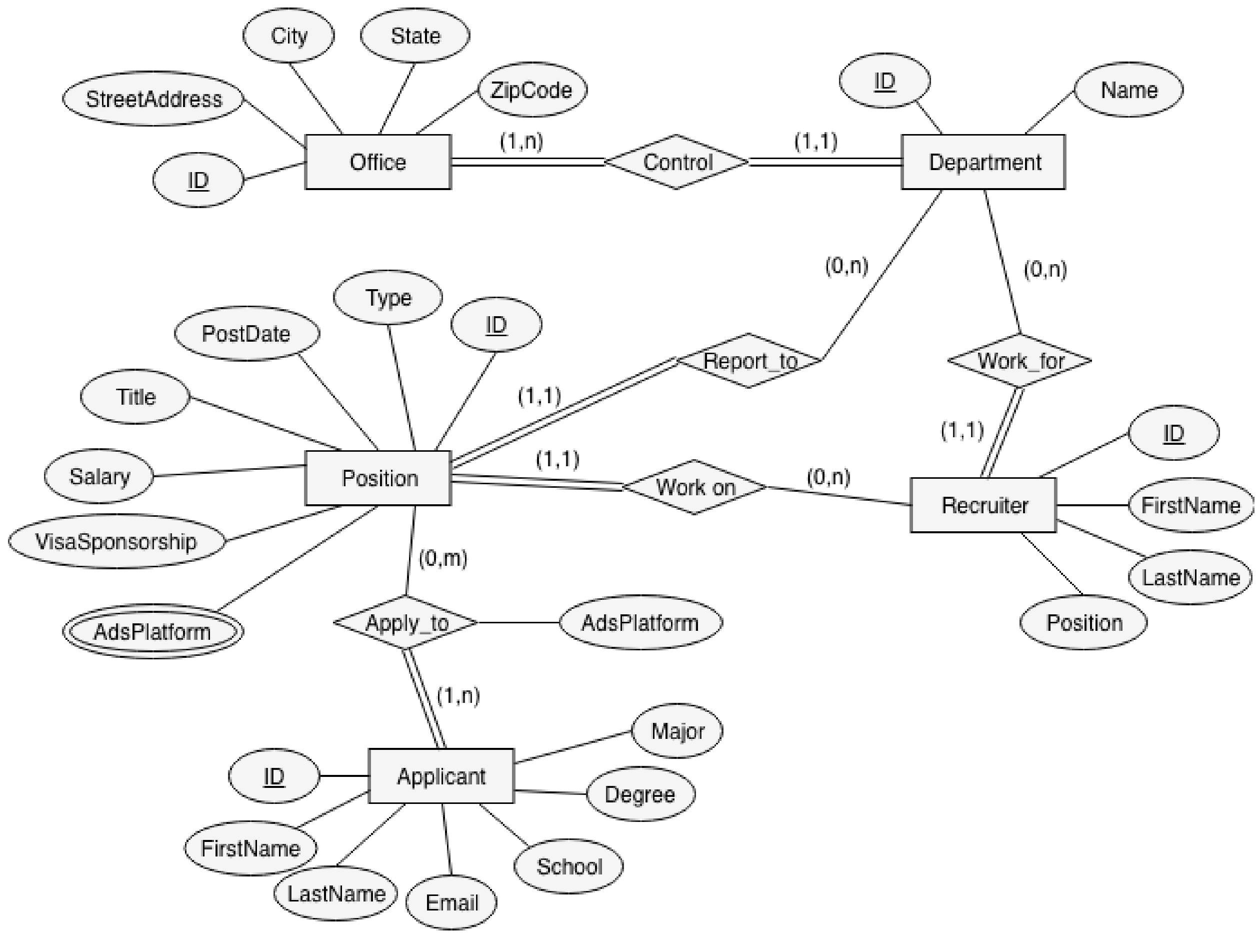


Figure 1: ER Diagram of the Job Application Database

There are five entities in our ER diagrams, including OFFICE, DEPARTMENT, POSITION, RECRUITER and APPLICANT. All of these entities have their own unique ID. The attributes of entity OFFICE are ID, Street Address, City, State and Zipcode. The attributes of DEPARTMENT are ID and department Name. The attributes of POSITION are ID, Type, PostDate, Title, Salary, VisaSponsorship and AdsPlatform. The attributes of RECRUITER are ID, FirstName, LastName and Position. The attributes of APPLICANT are ID, Email, FirstName, LastName, School, Degree and Major.

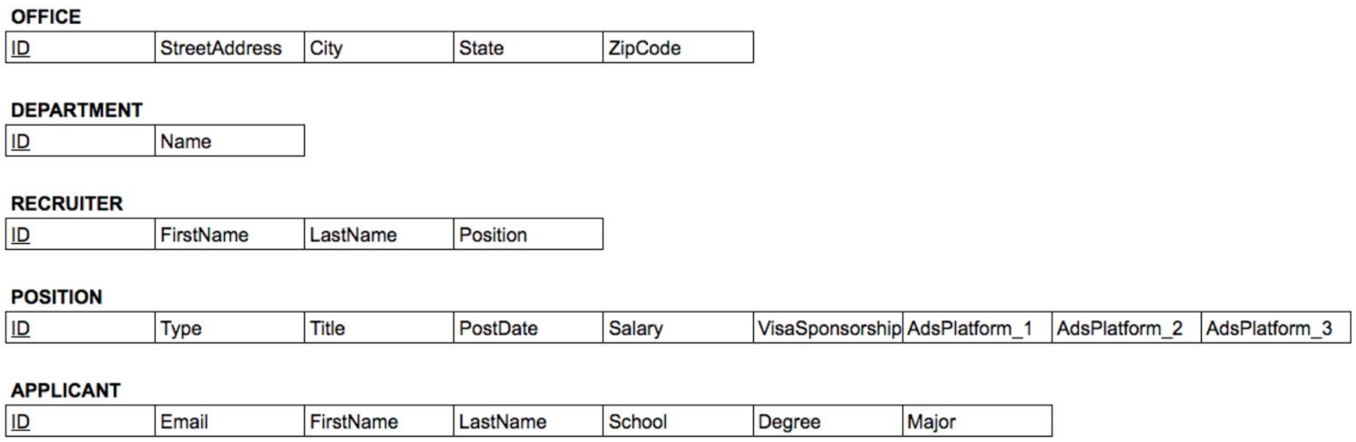
Each OFFICE controls one or more DEPARTMENTs and each DEPARTMENT must belongs to one and only one OFFICE. Each Recruiter works for one and only one DEPARTMENT. Each open POSITION will report to one and only one DEPARTMENT. However, a DEPARTMENT does not necessarily need to have RECRUITER (Sometimes, a department may borrow recruiter from other department to help recruit people on a position) or open POSITION. Each open POSITION must have one and only one RECRUITER working on it but a RECRUITER may not be responsible for any open POSITION temporarily. An APPLICANT can apply to one or more open POSITION and an open POSITION may have no APPLICANT.

There is no weak entities or relationships. Also, there is no class hierarchy in our ER diagram.

Task B: Relational Database Design Using ER-to-Relational Mapping

ER-to-Relational Mapping Algorithm

1. Step 1: Mapping of regular entity type. In this step, each of the strong entity in the ER schema is made into a relation with all the simple attributes. The ID attribute of each entity is chosen as the primary key. In this case, five relations are created as below in Figure 2.



OFFICE

ID	StreetAddress	City	State	ZipCode

DEPARTMENT

ID	Name

RECRUITER

ID	FirstName	LastName	Position

POSITION

ID	Type	Title	PostDate	Salary	VisaSponsorship	AdsPlatform_1	AdsPlatform_2	AdsPlatform_3

APPLICANT

ID	Email	FirstName	LastName	School	Degree	Major

Figure 2: Relations After Applying Step 1

2. Step 2: Mapping of weak entity type. This step is skipped because there are no weak entities in our ER model.
3. Step 3: Mapping of binary 1:1 relation type. It is skipped because there are no 1:1 relations in our ER model.
4. Step 4: Mapping of binary 1:N relation type. Include a foreign key in the N-side entity to reference to the other entity. In this case, a new attribute called “OfficeID” is added into the DEPARTMENT relation, which acts as a foreign key and references to the “ID” attribute in the OFFICE relation. A “DepartmentID” attribute is added into the RECRUITER relation as a foreign key to refer to the “ID” attribute in the DEPARTMENT relation. Following the same steps, “DepartmentID” and “RecruiterID” are added in the POSITION relation to refer to the IDs in the DEPARTMENT and RECRUITER relations. The modified relations are shown as below in Figure 3.

DEPARTMENT (DEPARTMENT.OfficeID -> OFFICE.ID)

ID	OfficeID	Name

RECRUITER (RECRUITER.DepartmentID -> DEPARTMENT.ID)

ID	FirstName	LastName	Position	DepartmentID

POSITION (POSITION.DepartmentID -> DEPARTMENT.ID; POSITION.RecruiterID -> RECRUITER.ID)

ID	Type	Title	DepartmentID	RecruiterID	PostDate	Salary	VisaSponsorship

Figure 3: Modified Relations After Applying Step 4

5. Step 5: Mapping of binary M:N relationship type. In our ER diagram, it is M:N relationship between the Position and Applicant entities, so their primary keys will form a new relationship called APPLY_TO and they act as foreign keys referencing to the IDs in the APPLICANT and POSITION relations. The attribute of the relationship “AdsPlatform” is added as an attribute in this relation as well. The relation is shown as below in Figure 4.

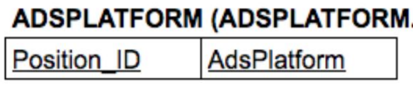
APPLY_TO (APPLY_TO.ApplicantID -> APPLICANT.ID; APPLY_TO.PositionID -> POSITION.ID)

ApplicantID	PositionID	AdsPlatform

Figure 4: Added APPLY_TO Relation After Applying Step 5

6. Step 6: Mapping of multivalued attributes. In our ER diagram, the “AdsPlatform” attached to the Position entity is a multivalued attribute, so a new relation called ADSPLATFORM

is made to map the “Position_ID” (foreign key that references to the POSITION relation) to the “AdsPlatform”. Since tuples may have duplicate entries of “Position_ID” and “AdsPlatform” attributes, they will form the primary key together. The relation is shown as below in Figure 5.



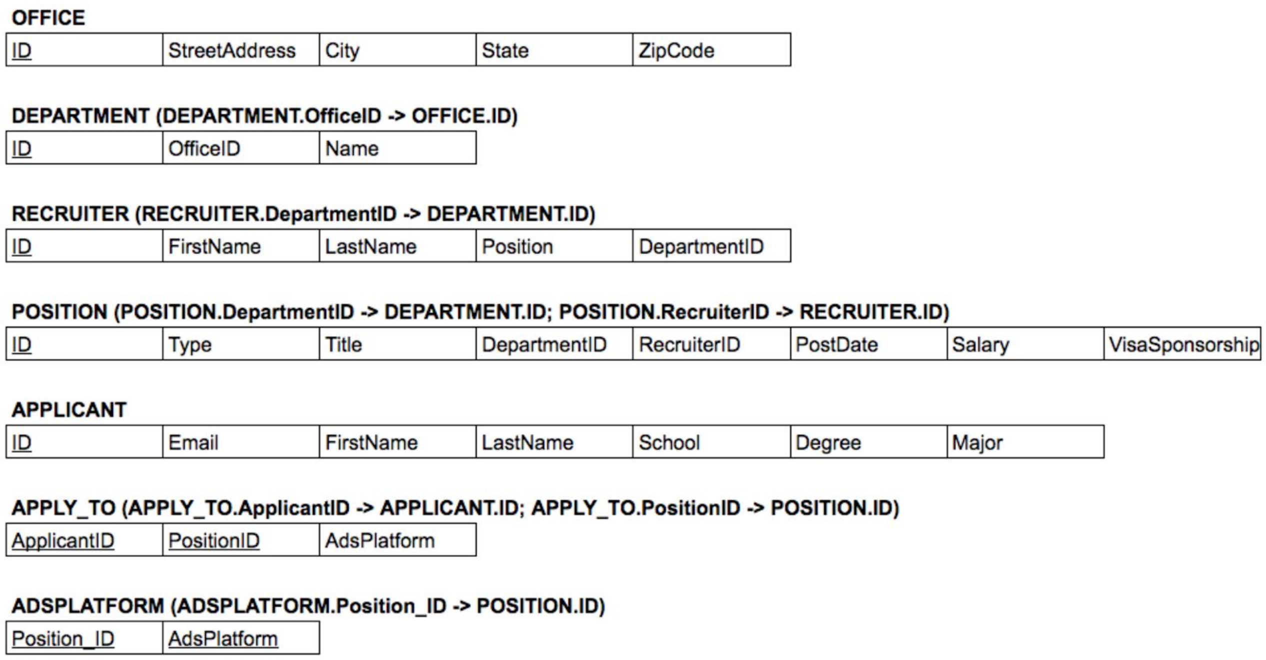
ADSPLATFORM (ADSPLATFORM.Position_ID -> POSITION.ID)

Position_ID	AdsPlatform

Figure 5: Added ADSPLATFORM Relation After Applying Step 6

7. Step 7: Mapping of N-ary relationship type. It is not applicable here, so it is skipped.
8. Step 8 and Step 9 are used for EER diagrams, so they are not applicable in our case.

Therefore, after all of steps, the relations are shown as below in Figure 6.



OFFICE

ID	StreetAddress	City	State	ZipCode

DEPARTMENT (DEPARTMENT.OfficeID -> OFFICE.ID)

ID	OfficeID	Name

RECRUITER (RECRUITER.DepartmentID -> DEPARTMENT.ID)

ID	FirstName	LastName	Position	DepartmentID

POSITION (POSITION.DepartmentID -> DEPARTMENT.ID; POSITION.RecruiterID -> RECRUITER.ID)

ID	Type	Title	DepartmentID	RecruiterID	PostDate	Salary	VisaSponsorship

APPLICANT

ID	Email	FirstName	LastName	School	Degree	Major

APPLY_TO (APPLY_TO.ApplicantID -> APPLICANT.ID; APPLY_TO.PositionID -> POSITION.ID)

ApplicantID	PositionID	AdsPlatform

ADSPLATFORM (ADSPLATFORM.Position_ID -> POSITION.ID)

Position_ID	AdsPlatform

Figure 6: Mapped Relations After Applying All the Steps

By putting this together using a arrow like Figure 9.2 in the textbook, the relationship will be like this in Figure 7 where the arrows show the directions that the foreign keys reference to:

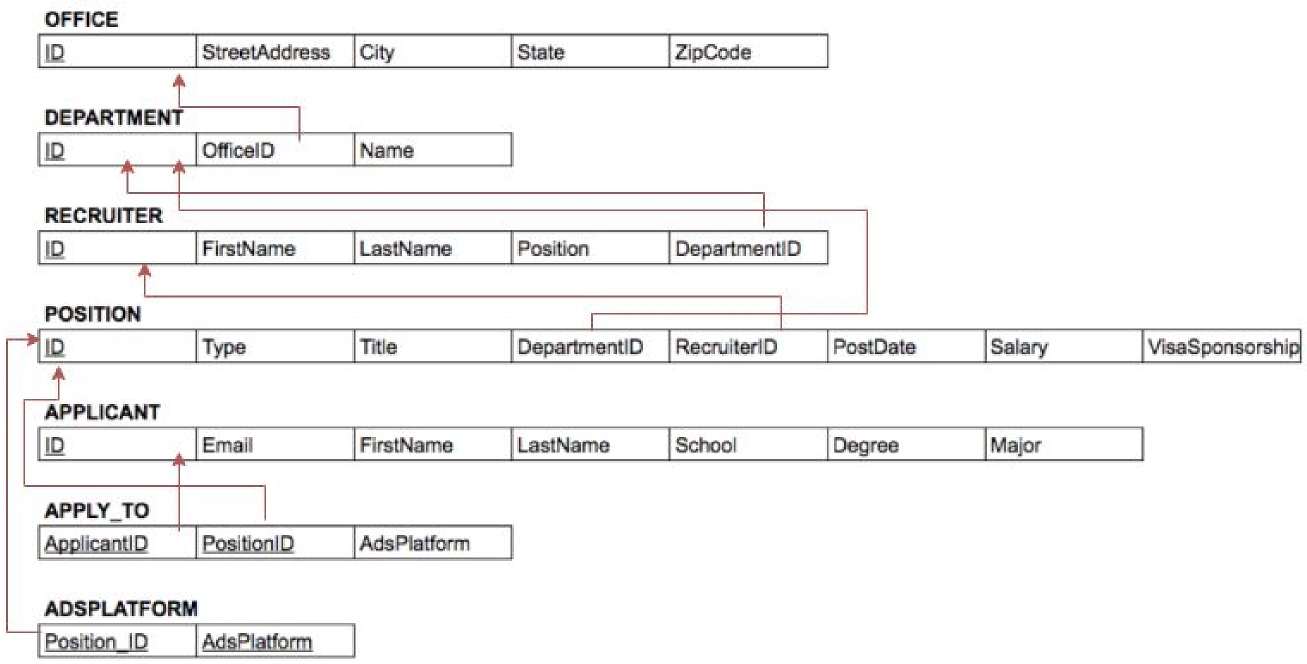


Figure 7: Mapped Relations Using Arrow Notations

To better summarize the mapping process, the following table as shown in Figure 8 lists the elements that construct the relations. The Office relation consists of the office entity only. The Department relation consists of the department entity and the control relationship. The Recruiter relation consists of the recruiter relationship and the work_for relationship. The Position relation consists of the position entity, report_to relationship and the works_on relationship. The Applicant relation consists of the applicant entity. The Apply_to relation consists of the apply_to relation. The AdsPlatform relation consists of the multi-valued attribute (AdsPlatform) from Position entity.

Relation Name	**ER Diagram components**
Office	E(Office)
Department	E(Department) + R(Control)
Recruiter	E(Recruiter) + R(Works_for)
Position	E(Position) + R(Report_to) + R(Works_on)
Applicant	E(Applicant)
Apply_To	R(Apply_to)
AdsPlatform	A(AdsPlatform)

Figure 8: Summary Table of Mapping

Schema of the database

OFFICE: Each office is uniquely identified with an office ID. We also believe no two distinct offices will have the same street address.

- Primary Key: ID
- Foreign Key: N/A
- Unique Keys: StreetAddress

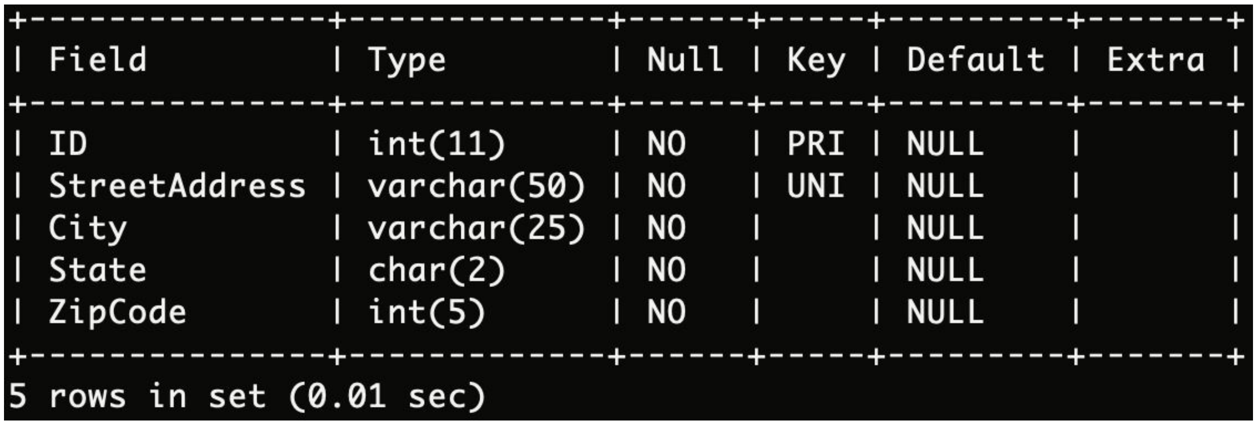
(1) **ID**: The unique identification number of an office. It has a datatype of INT and can only store up to 11 digits.

(2) **StreetAddress**: The name of street address of an office. It has datatype of VARCHAR and can only store up to 50 characters. The values of street address in this relation should be unique.

(3) **City**: The name of city where an office locates. It has a datatype of VARCHAR and can store up to 25 characters.

(4) **State**: The name of state where an office locates. It has a dataype of CHAR and can only store a string that contains 2 characters.

(5) **ZipCode**: The zip code of an office. It has a datatype of INT and can store up to 5 digits.



```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| ID            | int(11)     | NO   | PRI | NULL    |       |
| StreetAddress | varchar(50) | NO   | UNI | NULL    |       |
| City          | varchar(25) | NO   |     | NULL    |       |
| State         | char(2)     | NO   |     | NULL    |       |
| ZipCode       | int(5)      | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

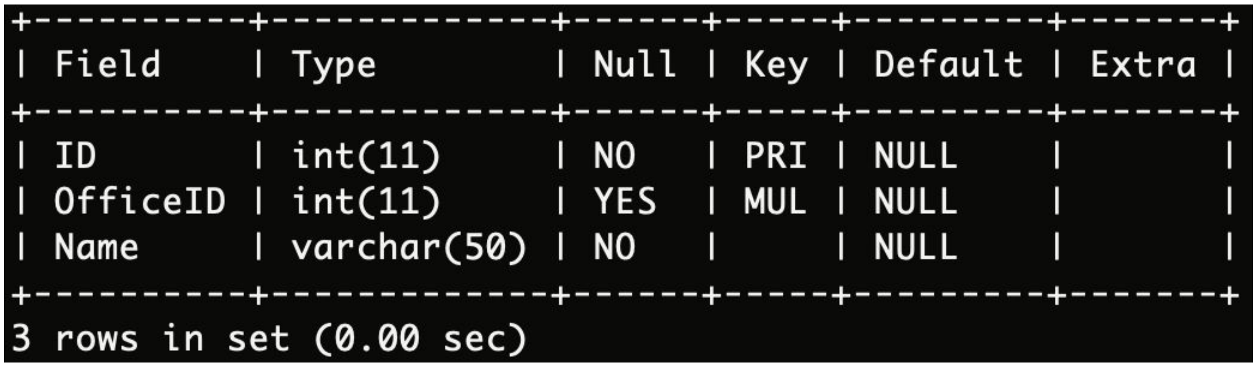
Figure 9. Relation Schema for OFFICE Relation

DEPARTMENT: Each department is uniquely identified with an department ID. Every department must belongs to an existing office. One office may have multiple departments. Also, different offices might have departments that have the same name (e.g. Both Seattle office and New York office have data science department)

- Primary Key: ID

- Foreign Key: OfficeID → Office.ID
- Unique Keys: N/A

(1) **ID**: The unique identification number of a department. It has a datatype of INT and can store up to 11 digits.
(2) **OfficeID**: The ID of the office where a department belongs. It has a datatype of INT and can store up to 11 digits. Since this is a foreign key, it is allowed to have NULL values. When the corresponding tuple for the referred table is deleted, the corresponding foreign key value will be set to NULL.
(3) **Name**: The name of department. It has a datatype of VARCHAR and can store up to 50 digits.



```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| ID       | int(11)     | NO   | PRI | NULL    |       |
| OfficeID | int(11)     | YES  | MUL | NULL    |       |
| Name     | varchar(50) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Figure 10: Relation Schema for DEPARTMENT Relation

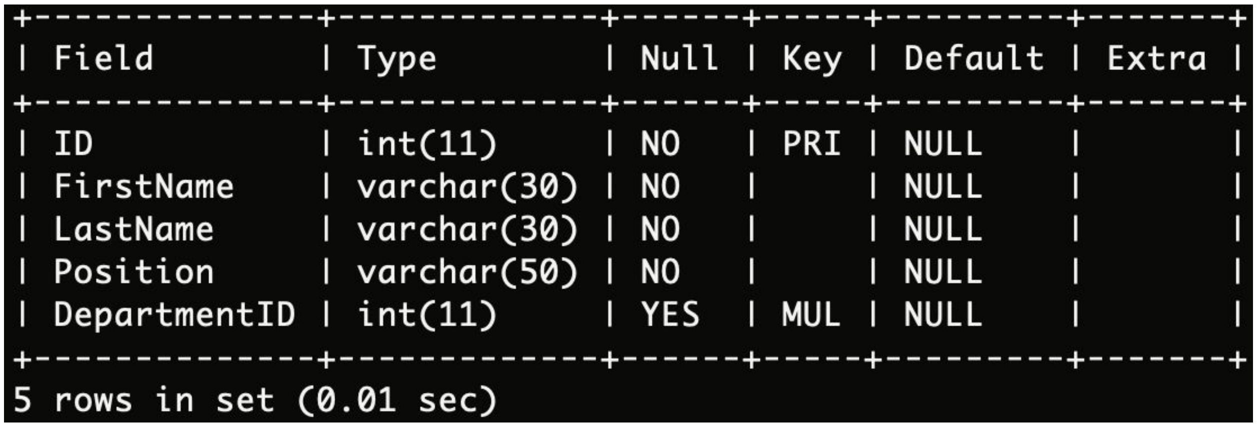
RECRUITER: Each recruiter is uniquely identified with an employee ID. Every recruiter must belong to an existing department

- Primary Key: ID
- Foreign Key: DepartmentID → Department.ID
- Unique Keys: N/A

(1) **ID**: The unique employee identification number of a recruiter. It has a data type of INT and can store up to 11 digits.
(2) **FirstName**: The first name of a recruiter. It has a data type of VARCHAR and can store up to 30 characters.
(3) **LastName**: The last name of a recruiter. It has a data type of VARCHAR and can store up to 30 characters.

(4) **Position**: The name of position that this recruiter is responsible for. It has a datatype of VARCHAR and can store up to 50 characters.

(5) **DepartmentID**: The ID of the department where a recruiter belongs. It has a datatype of INT and can store up to 11 digits. Since this is a foreign key, it is allowed to have NULL values. When the corresponding tuple for the referred table is deleted, the corresponding foreign key value will be set to NULL.



```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| ID          | int(11)     | NO   | PRI | NULL    |       |
| FirstName   | varchar(30) | NO   |     | NULL    |       |
| LastName    | varchar(30) | NO   |     | NULL    |       |
| Position    | varchar(50) | NO   |     | NULL    |       |
| DepartmentID | int(11)    | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

Figure 11: Relation Schema for RECRUITER Relation

POSITION_: Each open position is uniquely identified with a job ID. Every open position must belongs to an existing department and must be assigned to an existing recruiter

- Primary Key: ID
- Foreign Key: DepartmentID → Department.ID, RecruiterID → Recruiter.ID
- Unique Keys: N/A

(1) **ID**: The unique identification number of an opening position. It has a datatype of INT and can store up 11 digits.

(2) **Type**: The type of opening position (PartTime/FullTime). It has a datatype of CHAR and cao only store a string that contains 8 characters. Its default value is “FullTime” and it is allowed to have NULL value.

(3) **Title**: The name of an opening position. It has a datatype of VARCHAR and can store up to 50 characters.

(4) **DepartmentID**: The ID of the department that posted the position. It has a datatype of INT and can store up to 11 digits. Since this is a foreign key, it is allowed to have NULL

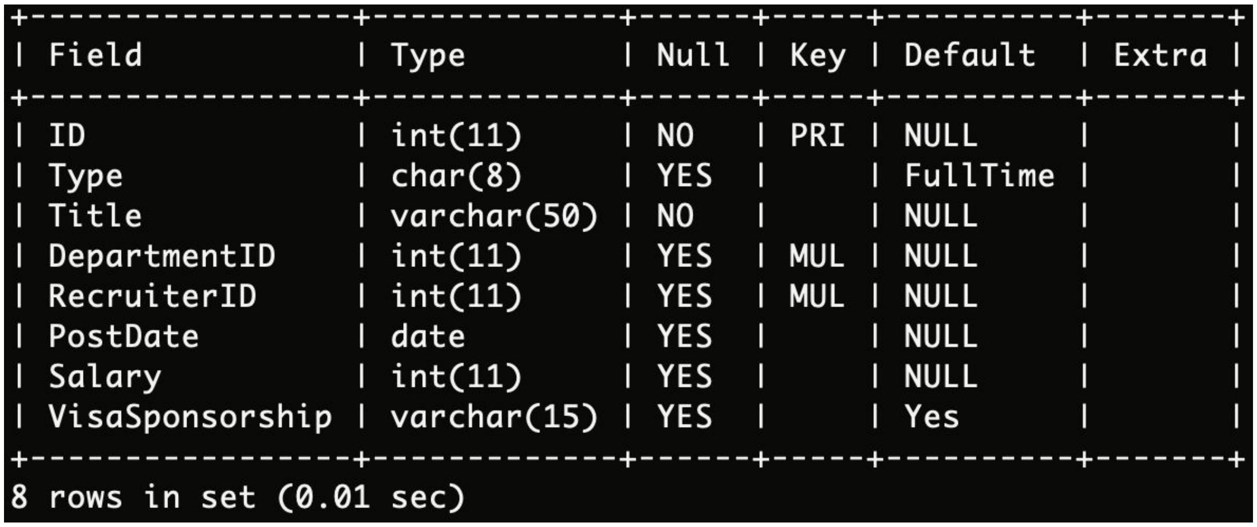
values. When the corresponding tuple for the referred table is deleted, the corresponding foreign key value will be set to NULL.

(5) **RecruiterID**: The ID of the recruiter who is responsible for the position. It has a datatype of INT and can store up to 11 digits. Since this is a foreign key, it is allowed to have NULL values. When the corresponding tuple for the referred table is deleted, the corresponding foreign key value will be set to NULL.

(6) **PostDate**: The post date of the position. It has a data type of DATE. It is allowed to have NULL values

(7) **Salary**: The annual salary ($) of the opening position. It has a datatype of INT and can store up to 11 digits. It is allowed to have NULL values.

(8) **VisaSponsorship**: The visa sponsorship of the position (YES/NO). It has a datatype of VARCHAR and can store up to 15 characters (allow short sentence for explaining unexpected situations). If it is not applicable, it is allowed to have NULL values.



```
+-----------------+------------+------+-----+----------+-------+
| Field           | Type       | Null | Key | Default  | Extra |
+-----------------+------------+------+-----+----------+-------+
| ID              | int(11)    | NO   | PRI | NULL     |       |
| Type            | char(8)    | YES  |     | FullTime |       |
| Title           | varchar(50)| NO   |     | NULL     |       |
| DepartmentID    | int(11)    | YES  | MUL | NULL     |       |
| RecruiterID     | int(11)    | YES  | MUL | NULL     |       |
| PostDate        | date       | YES  |     | NULL     |       |
| Salary          | int(11)    | YES  |     | NULL     |       |
| VisaSponsorship | varchar(15)| YES  |     | Yes      |       |
+-----------------+------------+------+-----+----------+-------+
8 rows in set (0.01 sec)
```

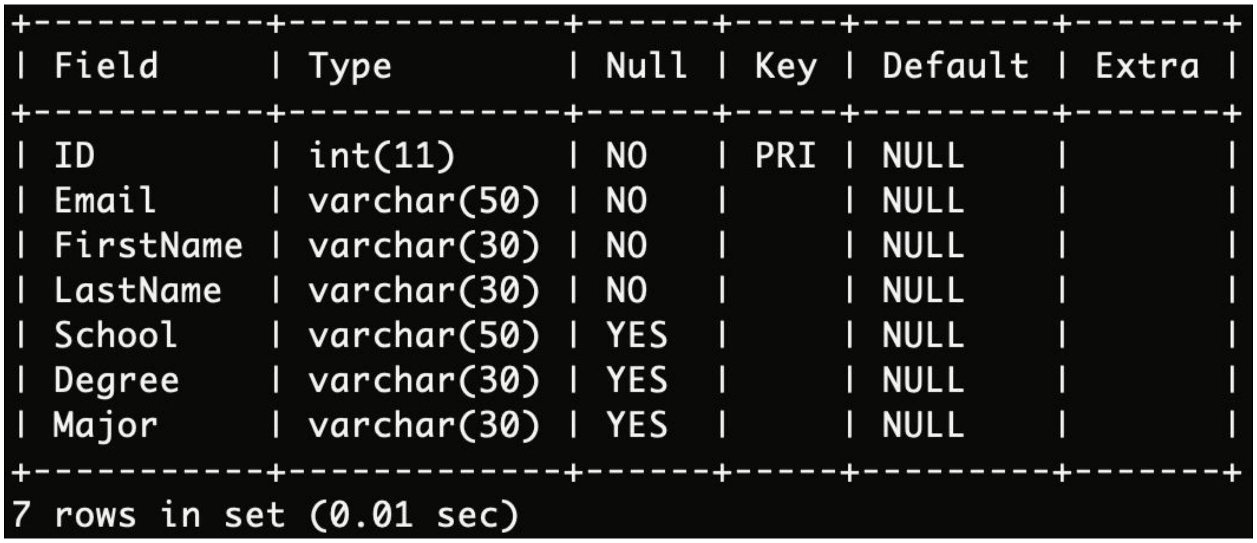
Figure 12: Relation Schema for POSITION_ Relation

APPLICANT: Each job applicant is uniquely identified with an applicant ID

- Primary Key: ID
- Foreign Key: N/A
- Unique Keys: N/A

(1) **ID**: The unique identification number of a job applicant. It has a datatype of INT and can store up to 11 digits.

(2) **Email**: The email address of a job applicant. It has a datatype of VARCHAR and can store up to 50 characters.
(3) **FirstName**: The first name of a job applicant. It has a datatype of VARCHAR and can store up to 30 characters.
(4) **LastName**: The last name of a job applicant. It has a datatype of VARCHAR and can store up to 30 characters.
(5) **School**: The name of school where the job applicant received his/her highest level of education. It has a datatype of VARCHAR and can store up to 50 digits. It is allowed to have NULL values.
(6) **Degree**: The name of degree that the job applicant received from his/her school. It has a datatype of VARCHAR and can store up to 30 characters. It is allowed to have NULL values.
(7) **Major**: The name of major that the job applicant studied in school. It has a datatype of VARCHAR and can store up to 30 characters. It is allowed to have NULL values



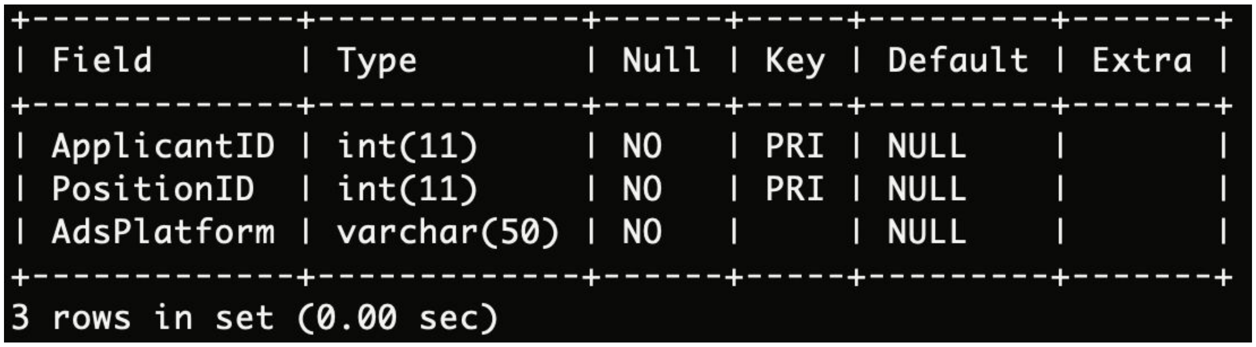
```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| ID        | int(11)     | NO   | PRI | NULL    |       |
| Email     | varchar(50) | NO   |     | NULL    |       |
| FirstName | varchar(30) | NO   |     | NULL    |       |
| LastName  | varchar(30) | NO   |     | NULL    |       |
| School    | varchar(50) | YES  |     | NULL    |       |
| Degree    | varchar(30) | YES  |     | NULL    |       |
| Major     | varchar(30) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

Figure 13: Relation Schema for APPLICANT Relation

APPLY_TO: Every registered applicant applies to an existing position through one or more of the platforms the company used for advertisement

- Primary Key: (ApplicantID, PositionID)
- Foreign Key: ApplicantID → Applicant.ID, PositionID → Position.ID
- Unique Keys: N/A

(1) **ApplicantID**: The unique identification number of a job applicant. It has a datatype of INT and can store up to 11 digits.
(2) **PositionID**: The ID of position for which the applicant applied. It has a datatype of INT and can store up to 11 digits.
(3) **AdsPlatform**: The name of platform through which the applicant applied. It has a datatype of VARCHAR and can store up to 50 characters.



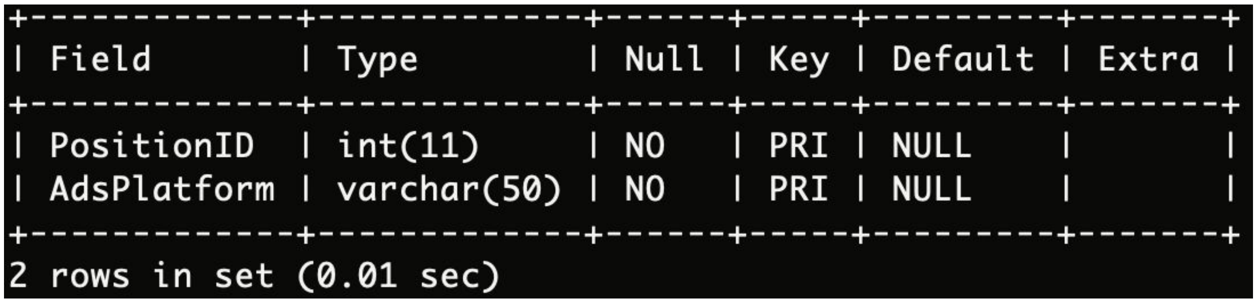
```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| ApplicantID | int(11)     | NO   | PRI | NULL    |       |
| PositionID  | int(11)     | NO   | PRI | NULL    |       |
| AdsPlatform | varchar(50) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Figure 14. Relation Schema for APPLY_TO Relation

ADSPLATFORM: Each existing position is advertised on one or more platforms

- Primary Key: (PositionID, AdsPlatform)
- Foreign Key: PositionID → Position.ID
- Unique Keys: N/A

(1) **PositionID**: The unique identification number of an opening position. It has a datatype of INT and can store up to 11 digits.
(2) **AdsPlatform**: The name of platform on which the company advertised an opening position. It has a datatype of VARCHAR and can store up to 50 characters.



```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| PositionID  | int(11)     | NO   | PRI | NULL    |       |
| AdsPlatform | varchar(50) | NO   | PRI | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

Figure 15. Relation Schema for ADSPLATFORM Relation